```
/*
Display.h - A simple track GPS to SD card logger. Display module.
TinyTrackGPS v0.13

Copyright © 2019-2021 Francisco Rafael Reyes Carmona.
All rights reserved.

rafael.reyes.carmona@gmail.com

  This file is part of TinyTrackGPS.

  TinyTrackGPS is free software: you can redistribute it and/or modify
  it under the terms of the GNU General Public License as published by
  the Free Software Foundation, either version 3 of the License, or
  (at your option) any later version.

  TinyTrackGPS is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
  GNU General Public License for more details.

  You should have received a copy of the GNU General Public License
  along with TinyTrackGPS.  If not, see <https://www.gnu.org/licenses/>.
*/

#if ARDUINO >= 100
  #include "Arduino.h"
#else
  #include "WProgram.h"
#endif

#ifndef Display_h
#define Display_h

#include "config.h"

#if defined(DISPLAY_TYPE_LCD_16X2)
    #include <LiquidCrystal.h>
#elif defined(DISPLAY_TYPE_LCD_16X2_I2C)
    #include <LiquidCrystal_I2C.h>
#elif defined(DISPLAY_TYPE_SDD1306_128X64) || defined(DISPLAY_TYPE_SH1106_128X64)
    #define U8X8_HAVE_HW_I2C
    #include <U8x8lib.h>
    //#include <U8g2lib.h>
#elif defined(DISPLAY_TYPE_SDD1306_128X64_lcdgfx)
    #include <lcdgfx.h>
    #include <lcdgfx_gui.h>
#endif

enum Display_Type {
    SDD1306_128X64,     // Para usar pantalla OLED 0.96" I2C 128x64 pixels
    LCD_16X2,           // Para usar LCD 16 x 2 carateres.
    LCD_16X2_I2C        // Para usar LCD 16 x 2 carateres. I2C.
};

class Display {
    private:
        //byte _offset;
        byte _width;       // Width pixels or numbers of columns for LCD.
```

```
60          byte _height;       // Height pixels os numbers of rows for LCD.
61          Display_Type _screen;
62          #if defined(DISPLAY_TYPE_LCD_16X2)
63              LiquidCrystal* lcd;
64          #elif defined(DISPLAY_TYPE_LCD_16X2_I2C)
65              LiquidCrystal_I2C* lcd;
66          #elif defined(DISPLAY_TYPE_SDD1306_128X64)
67              //U8G2_SSD1306_128X64_NONAME_1_HW_I2C* u8g2_SSD1306;
68              U8X8_SSD1306_128X64_NONAME_HW_I2C* u8x8_SSD1306;
69          #elif defined(DISPLAY_TYPE_SH1106_128X64)
70              U8X8_SH1106_128X64_NONAME_HW_I2C* u8x8_SH1106;
71          #elif defined(DISPLAY_TYPE_SDD1306_128X64_lcdgfx)
72              DisplaySSD1306_128x64_I2C* display;
73          #elif defined(DISPLAY_TYPE_HX1230_96X68)
74              U8G2_HX1230_96X68_1_3W_SW_SPI* u8g2_HX1230;
75          #endif
76
77      public:
78          Display(Display_Type t = SDD1306_128X64);
79          Display() = delete;                             // Constructor por defecto.
80          Display(const Display&) = delete;               // Constructor de copia.
81
82          void start();
83          void clr();
84          void print(int, int, const char[]);
85          void print(int, const char[]);
86          void print(const char[]);
87          void print(const char[], const char[]);
88          void print(const char[], const char[], const char[]);
89          void print(const char[], const char[], const char[], const char[]);
90          void wait_anin(unsigned int);
91          void draw_wait(byte);
92          void print_PChar(byte);
93          void DrawLogo();
94          void drawbattery(uint8_t);
95          Display_Type display_type(){return _screen;};
96  };
97
98  extern const uint8_t TinyTrackGPS_font8x16[] PROGMEM;
99
100 #endif
```